RECEIVED
CENTRAL FAX CENTER
JAN 0 6 2005

AMENDMENTS TO THE SPECIFICATION:

Please amend the paragraph beginning on page 2, line 4 as follows:

--Previous disclosed techniques for storing and playing data based on using the PES, or Packetized Elementary Stream, format which allows for efficient content movement is described in commonly-owned, co-pending U.S. Patent Application Nos. 09/534643[[,]] and 09/535069, and in issued U.S. Patent No. 6,662,329 ~~09/535001~~ the contents and disclosure of each of which are incorporated by reference herein.--

Please amend the paragraph beginning at page 7, line 7 as follows:

-- According to the invention, there is provided a technique for inserting, or remultiplexing, packets containing new content with a filtered transport stream as it is being stored to memory for subsequent filing on a fixed storage device. The technique is based on a modification to the existing transport demultiplexor as described in commonly-owned, ~~co-pending~~ United States Patent ~~Application Serial~~ No. 6,275,507 ~~08/938,248~~ entitled TRANSPORT DEMULTIPLEXOR FOR AN MPEG-2 COMPLIANT DATA STREAM the contents and disclosure of which is incorporated by reference as if fully set forth herein.--

Please amend the paragraph at page 7, beginning at line 15 as follows:

-- Particularly, the demultiplexor described in United States Patent ~~Application Serial~~ No. 6,275,507 ~~08/938,248~~ is a transport demultiplexor that is adapted for demultiplexing an MPEG-2-compliant transport stream into system data streams, a video data stream, and an audio data stream particularly, by extracting program clock references (PCRs) from the data stream and filtering out unnecessary components through the use of Packet Ids (PIDs). As shown in Figure 1, the transport demultiplexor 10 includes front end logic 15, back-end logic 20, and including a packet buffer 21, control circuit 25, and a data unloader 26, video unloader 27, and audio unloader 28. Generally, the front end logic 15 receives transport stream input packets, and delivers the transport stream packets to the packet buffer 21. The packet buffer 21, in turn, delivers system data to the system data unloader 28, video data to the video unloader 26, and audio data to the audio unloader 27 each of which, as will be explained, asynchronously pull packets out of the packet. --

Please amend the paragraph at page 9, beginning at line 8 as follows:

-- As described in commonly-owned, co-pending U.S. Patent Application Serial No. [[_____]] 09/730,636 entitled SYSTEM AND METHOD FOR REMULTIPLEXING OF A FILTERED TRANSPORT STREAM ~~[END920000140US1, Atty Dckt. #13950]~~, the output stream is additionally input to a packet loader device 18 which may transport the packets through the buffer control device 25 for loading into a packet buffer 21, which may be a ten (10) packet bucket, for example. As mentioned, the PID filter 14 enables retention of only the packets of

interest however, this data has not been separated. Preferably, a key word or an information word has been is associated with each packet which identifies the packet as either audio, video or data headed for system memory. Thus, loaded into the packet buffer are all the packets of interest with each packet having an information word appended thereto indicating the payload and dictating the subsequent processing to be performed by the video, audio and data unloaders. This processing includes transferring packets associated with one program, e.g., audio data, video data and navigation/system data associated only with that program, in a "bucket queue" memory for subsequent access including decoding and playback. That is, the special controls in the data loader 28 and the buffer control 25 enable the video, audio and system data to be stored together in one place, the bucket queue, rather than separate places. More particularly, as described in co-pending U.S. Patent Application Serial No. [[_____]] 09/730,636, the transport demultiplexor 10 of Figure 1, provides a queue remux component 100 which performs a packet insertion function enabling new data content to be subsequently inserted for storage in the bucket queue that was not in the original stream. As shown in Figure 1, this queue remux component 100 is provided as part of the data unloader module 28.--